



*Science Enhanced
Network Domains and
Secure Social Spaces*

SENDs

SENDSim

A modeling and simulation software tool for understanding network security threats and scientifically experimenting with policy and human solutions.

Software Specification

Version 1.0

PREPARED FOR DIRECTED TECHNOLOGIES, INC.

BY ICOSYSTEM CORPORATION

JUNE 17, 2011

1 Contents

Executive overview	3
1 Introduction	4
1.1 Caveats	4
1.2 Background.....	4
1.3 Modeling approach	6
2 Software	7
2.1 Architecture.....	7
2.2 Input levers.....	7
2.3 Output metrics and visualizations	10
2.4 Utilities.....	13
3 Model	14
3.1 Computer network	15
3.2 Malware.....	16
3.3 Computer.....	18
3.4 Worker.....	19
3.5 IT administrators	24
3.6 Vendor	24
3.7 Hacker.....	24
3.8 Insider threat	25
4 Example questions and metrics	25

Executive overview

Cybersecurity professionals face a daunting mix of ever-changing threats and a technological landscape that grows in complexity every day. Preparing a robust plan for network defense requires not just a detailed understanding of information technology but a careful appreciation of how the actions and inactions of ordinary users can enhance or compromise security. Some security plans may on the surface seem robust but can be circumvented by users in their quest to accomplish work tasks. Other plans may meet security and information assurance goals but inadvertently prevent legitimate operations from running effectively. Further, even the best plans and designs cannot be executed unless security experts can motivate institutional leaders and decision makers to appreciate the real threats of cybersecurity in an environment of increasingly tight budgets and conflicting priorities.

SENDSim helps cybersecurity professionals face these challenges by providing a platform for understanding threats, predicting user actions, evaluating solutions, and communicating the benefits of a principled security plan to non-technical decision makers. Users can specify network designs, assumptions, and policy parameters. SENDSim then creates a simulated network, a simulated workforce using that network, and a simulated malware threat. Incorporating modeling techniques from epidemiology and behavioral economics, SENDSim captures both the behaviors of the malware and the behaviors of the network users. This includes users' appreciation of cyber threats, their level of technical sophistication, and their actions, such as choosing task strategies, enabling and disabling features, and telling co-workers about threats and solutions. Detailed visualizations demonstrate how malware infiltrates a network, spreads, and inflicts damage. In addition to standard cybersecurity metrics focused on technology, SENDSim adds metrics related to the workforce, such as productivity, as well as metrics related to cost. Users can then perform *in silico* experiments with different strategies and evaluate possible outcomes without the cost and risk of experimenting with real networks.

By modeling both the behavior of malware and the behavior of non-malicious users, the cybersecurity professional can take a broader view of how their designs and policies impact both their security goals and the larger productivity and efficacy goals of the institution. This broader view will also aid in their communication to non-technical decision makers. Cybersecurity professionals are often charged with assessing the vulnerabilities of a network and recommending a course of action. SENDSim will help these experts make cyber threats more tangible for decision makers, helping them visualize possible threat scenarios and quantify those scenarios in terms of dollars-and-cents cost as well as in terms of information assurance and military readiness. Decision makers can then draw more informed conclusions that take the broader interests of the institution

into account. Future versions of SENDSim may include a module for prescribing an optimal strategy to meet user-specified network, budget, and security goals.

1 Introduction

1.1 Caveats

This specification is a living document; it will be continuously updated as details become fleshed out and new information and feedback become available. Its purpose is to aid in communication between the SENDSim development team and the larger SENDS community. Nothing in here is set in stone, and details can and will change over the course of the project. This document is neither a contract nor is it making any commitments as to what will and won't be accomplished in any particular phase of effort, even though it sometimes uses commitment-like phrasing. We shouldn't try to use this document as a means for scoping and committing; rather, it should be a means of communicating the direction in which we want to go. From there, we can implement pieces in order according to their priority and value to the end user.

For clarification, elements not implemented in the first phase of work (ending June, 2011) are shaded in a light tan.

1.2 Background

Cyberspace, although it is the result of tremendous technological progress, is not just a piece of technology: it is both an enabler and an amplifier of human nature. It feeds (and in many ways feeds on) one of the most fundamental needs of human beings: communication. That it has become such an integral part of our lives in such a short time shows how deeply it resonates with our need to communicate and to be connected. It should come as no surprise, then, that the multifaceted dynamics of cyberspace be so strongly influenced, even defined, by the behavior of its participants. According to Mark Graff of Lawrence Livermore National Laboratory: [cyberspace] gives individuals and small groups unprecedented reach to affect others; it makes physical distance much less of an insulating factor; confuses us about what is permanent, or public, or safe; and largely operates insensibly to us. We feel safer if important data is near us, or some place we know, or with someone we've met, but these comfort factors make no "Internet" sense and don't scale to Internet dimensions. In matters of risk assessment, we feel pretty safe from attacks originating "far away;" we also tend to ignore "low and slow"—or sporadic—attacks; random, "pointless" attacks (like from Internet worms) mostly tend to be low on our worry list, too. No wonder that the intuition we have gained from the physical world over thousands of years of evolution leaves us ill prepared to deal with the new geography of cyberspace. We can't hope to

acquire this new kind of intuition overnight. The bad news is that we suffer from severe limitations in our understanding of a critical component of our lives. The good news is that we are all subject to the same limitations—good news only if we can regain a competitive advantage in what has been a level playing field. Understanding our own behavior and that of our enemies becomes the most viable defense and the most potent weapon we can develop.

In this context, we believe that modeling and simulation (M&S) with a focus on human behavior will provide the unifying framework for designing and experimenting with new concepts and methods, borrowing from multiple disciplines such as psychology, economics (e.g., incentives, game theory), behavioral economics (e.g., decision heuristics and cognitive biases), social sciences and more. A survey of M&S in the field of Network Defense and Security (NDS) reveals the field's quasi exclusive focus on the technical aspects of security (physical devices, e.g., routers, servers, computer hosts, mobile devices; intrusion detection; protocols; encryption; network sensors) with the aim of replicating the properties of worms, viruses, intrusions, distributed denial of service attacks (DDoS), and testing technical means of preventing, detecting and responding to these threats. Such efforts are essential for improving the technical aspects of cybersecurity and significant investments need to be made to ensure continuous progress—and to keep up with increasingly sophisticated enemies. At the same time, human behavior has been consistently overlooked as a source of security weaknesses and has been completely ignored as a possible emergent source of security *robustness*. More precisely, although there is an almost universal recognition in the field of NDS that human behavior is the weakest link in security, it has not been studied, not to mention exploited, to the degree it deserves. In fact, the “bad guys” have been exploiting the weaknesses of human cognition much more effectively than we even begin to understand. However, we should feel fortunate that the weakest component of security is also one of the most stable: whereas technology evolves at breakneck speed, the fundamentals of human nature remain unchanged, even as they are constantly finding new avenues for expressing themselves. The complexity of securing cyberspace is staggering, and so is even the more mundane problem of NDS. The only constant we can hope to leverage to reduce this *wicked problem* to a simpler form is human behavior.

But the technology aspects of NDS are probably easier to define and tackle, and technology-based NDS solutions seem, at first glance, to be more commercially viable than behavioral interventions. As a result, the entire field is akin to the drunken man looking for his car keys right under the street light—not because he thinks that's where they are, but because that's where he can see. As an illustration, it seems surreal that in an otherwise excellent document, the authors of a 2009 manifesto from Sandia National Laboratories entitled “Complexity Science Challenges in Cybersecurity” have not dedicated a single line to human behavior (Armstrong et al., 2009). For example, their main M&S thrust is entitled: “Modeling the behavior of programs, machines, and

networks”. No humans necessary. We concur with the authors that there is a need for a new “cyber-calculus”—just the ability to frame concepts and issues in modern mathematical terms would be of enormous help. However, there is a piece missing that might be responsible for the vast majority of our NDS nightmares (which could come true any moment) and would offer the largest ROI if it were understood from a scientific perspective: human behavior, not just at the interface between man and machine (user interfaces and system interaction), but at the more fundamental level of the cognitive heuristics humans use to evaluate trust and distrust and to make all sorts of other decisions (Tversky and Kahneman, 1986).

1.3 Modeling approach

A typical complex systems model replicates a real-world system, like a market, an organization, or a particular process in a software tool that can be used to understand the behavior of that system, make predictions about the future, and test the implications of various actions. The underlying technology behind the SENDSim model is called Agent-based Modeling (ABM), which simulates the real world by creating software environments composed of many interacting autonomous modules or artificially intelligent agents. Each agent makes decisions and acts based on a set of properties and rules.

To build an ABM, one must:

- a) Identify the output metrics that are pertinent,
- b) Define the properties and rules of the main actors in the system as they relate to the metrics, and
- c) Define the interactions between the actors.

The model has “levers,” or exposed variables and assumptions that the user can manipulate to set up different “what-if scenarios.” These inputs feed into the model and in conjunction to the rules and properties of the agents determine the results. An optimizer can be included to test thousands of lever permutations to find a business strategy or a system configuration that maximizes the objectives and satisfies the constraints of the simulated environment. The figure below provides a high-level schema of the model.



2 Software

This section specifies all the components of SENDSim that are related to the technical architecture and utility, rather than the details of the simulation model itself.

2.1 Architecture

2.1.1 Technical

SENDSim is developed in Java using Icosystem's proprietary modeling and simulation libraries. It will be deployed via a password-protected Web server for access by the SENDS community.

2.1.2 User experience

The SENDSim user interface facilitates rapid exploration and comparison of threat scenarios under different policy, network, and workforce assumptions. Users specify input lever settings (or use the defaults), run simulations, and then view simulation results. More details are provided in the SENDSim User Manual and Release Notes.

2.2 Input levers

Default values are in parentheses

2.2.1 People

1. Number of workers (250)
2. Worker labor cost (\$50,000)
3. Number of IT administrators (10)
4. IT administrator labor cost (\$65,000)
5. Number of workgroups (10)
6. Standard deviation of workgroup size (12.5)
7. Inter-network connectivity of workers
8. Initial mean threat appreciation (0.286)

9. Peer pressure strength (0.5)
10. Initial mean technical knowledge (0.45)
11. Infection discovery
 - a. Likelihood that an infection is reported to administrative staff (0.75)
 - b. Average number of coworkers told (2)
12. Workstation management

2.2.2 Computer network

1. Workstation types
 - a. Convenient means to describe workstation configurations and vulnerabilities, by archetypes, e.g., a “New” workstation with Win7 and autoupdate enabled, an “Old” workstation with WinXPsp1 and autoupdate off).
2. Presets (selection of a preset will populate levers listed below with tbd predefined values)
 - a. Mature network – Initialized with a variety of workstation ages, operating systems, and vulnerabilities.
 - b. New implementation – All initialized with the same, specified, properties
 - c. Classified – All initialized with properties consistent with SIPRNet workstations
3. Minutes to fix an infection workstation (120)
4. Number of SCADA devices attached to the network (0)
5. Cost of SCADA device damage (\$1,000)
6. Computer vulnerabilities
 - a. NetBIOS Buffer Overrun MS08-067 (100)
 - b. Removable media (250)
 - c. Removable media with AutoRun (250)
 - d. LNK/PIF MS10-046 (100)
 - e. Printer Spooler Service MS10-06 (100)
 - f. Server RPC MS08-067
 - g. ADMIN\$

2.2.3 Security policy

1. Technology enforcement
 - a. Push new defaults to existing workstations (w/wo opt-out, opt-in)
 - b. Features enabled/disabled by default
 - i. Removable media (enabled by default)
 - ii. Autorun (enabled by default)
 - iii. Windows file sharing (enabled by default)
 - iv. Windows file sharing across workgroups (disabled by default)

- c. Fraction of workers with admin permissions to their own workstation (assume correlated with tech sophistication)
- d. File permissions
 - i. Everyone has full read access
 - ii. Read access only to files related to current mission
 - iii. Read access only to files within workgroup
 - iv. Everyone has read access to an unclassified abstract of every file
- 2. Human enforcement
 - a. Features allowed/disallowed
 - i. Removable media
 - ii. Personal devices allowed/disallowed
 - b. No-fault event reporting
 - c. Advertized penalties for policy violation
 - d. Education campaigns
 - i. Threat awareness
 - ii. General technical proficiency
 - e. Deploy fake malware
 - f. Preemptively identify high-risk workers
- 3. Distinction being CD-R policy and USB-drive policy
- 4. Data exfiltration detection
 - a. Number of files accessed by a single user in a period of time before alarm is triggered.
- 5. Countermeasures during attack
- 6. Presets
 - a. No strategy
 - b. Full lock-down
 - c. Empowered users
 - d. More admins

2.2.4 Attack scenarios

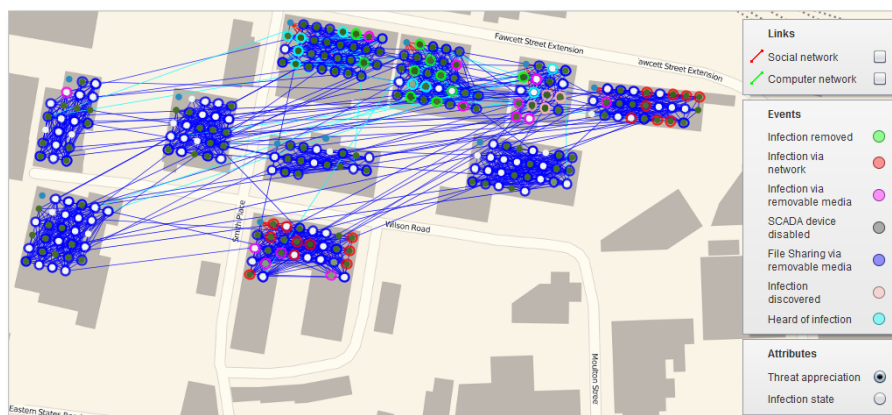
- 1. Number of days simulated (5)
- 2. Randomization seed (6)
- 3. Conficker (enabled by default)
 - a. Likelihood of discovery (100%)
 - b. Currently unrealized propagation and attack mechanisms
 - i. Removable-media propagation without requiring AutoRun (disabled by default)

- ii. Zero-day autonomous propagation mechanism
 - iii. Data Exfiltration
 - iv. Denial-of-service
 - v. Workstation software destruction
- 4. Stuxnet (disabled by default)
 - a. Likelihood of discovery (100%)
 - b. Currently unrealized propagation and attack mechanisms
 - i. PLC Rootkit with zero-day modification
- 5. Insider threat (disabled by default)
 - a. Single threat or network of threats
- 6.

2.3 Output metrics and visualizations

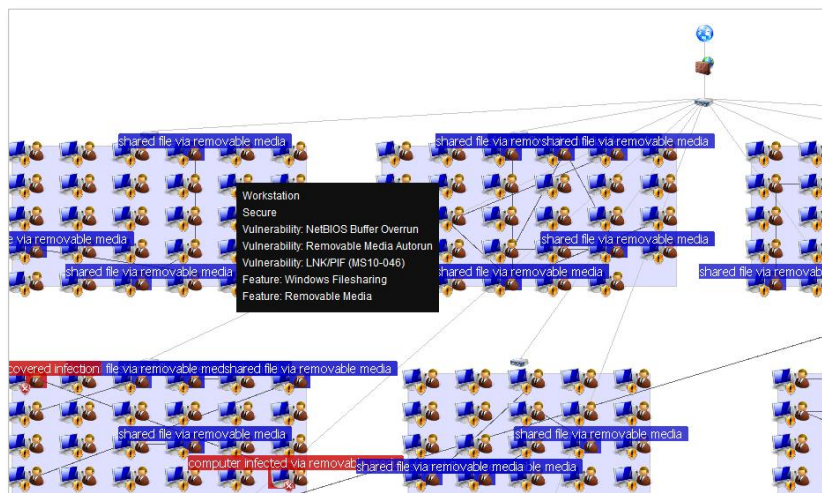
2.3.1 Map visualization

The computer network and social network are displayed on a geographical map. Each workgroup is represented as a building. Network links and events are overlaid.



2.3.2 Network topology visualization

The computer and social networks are combined in a fixed topology organized by workgroup.



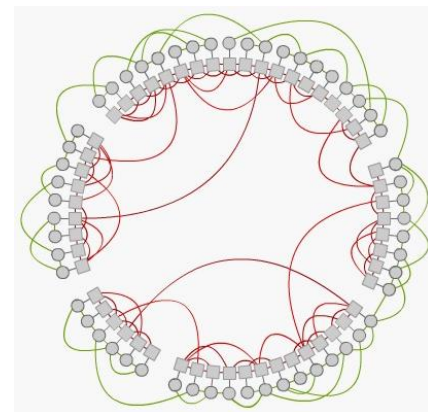
1. Node icons and edges change to reflect events as they occur.
2. “Thought” bubbles appear over a node indicating events happening at that node at a particular point during the simulation, e.g., “computer infected” or “computer repaired”.
3. Clicking on a node pop up detail about that node.
4. The user can drag the visualization around and zoom in on particular sections.

2.3.3 Social network visualization

A circle graphs is used to separate out fully connected intra-workgroup links from sparsely connected inter-workgroup links.

2.3.4 Time series

Some quantitative output metrics will be displayed as time series charts (i.e., a line graph of the metric with time on the x-axis) that fill in as the simulation progresses. Some examples:



Technical

1. Infection density
2. Vulnerability density
3. Information security
4. Network availability

Behavioral

1. Worker productivity
 - a. Worker agents have tasks to accomplish every day. Tasks are weighed by importance. Every time a task is accomplished, its weight adds to the total productivity metric. Thus, the absolute value of the productivity metric is not particularly useful, but relative differences in productivity, either over time or between scenarios, may prove to be useful.
2. Mean technical knowledge
 - a. Each worker will have a level of technical knowledge that increases with education.
3. Mean threat appreciation
 - a. See Section 3.4.3: Threat appreciation.
4. Maximum number of files accessed by a single person
5. Total number of secret files exfiltrated

Other

1. Cost
 - a. Lost productivity
 - b. IT administration overhead
2. Functional Performance (aka, "utility")
 - a. A metric that represents a weighted combination of all the network goals and values.
 - b. Users will be able to specify how much each other metric contributes to this metric.
 - c. For example, a network for researchers that contains highly classified data may have a utility metric that is weighted heavily in favor of information security and less toward productivity. Similarly, a network for war-fighting operations might weigh readiness and productivity above information security.

2.3.5 Non-timeseries metrics

1. SIR parameters
 - a. The susceptible-infected-recovered (SIR) model is a classic dynamical systems representation of epidemiological time series. While simple, it offers useful analytical mechanisms for predicting how fast an infection will spread and whether it will slowly become an epidemic or quickly die off.

- b. The SIR model is typically described as a set of three differential equations:

$$\frac{dS}{dt} = -\beta IS \quad \frac{dI}{dt} = \beta IS - \nu I \quad \frac{dR}{dt} = \nu I$$

where S is the fraction of the population that is susceptible to infection, I is the fraction infected, and R is the fraction recovered and immune to reinfection. The constants β and ν define the rate of infection and the rate of recovery.

- c. In the SIR model, these constants are *inputs* that uniquely define the output of the model. In SENDSim, we may find it helpful to quantify the timescale of a simulated malware outbreak using an approximation of β and ν as model *output* metrics. To say that a simulation has a certain β and ν is to say that, if one were to approximate the outbreak using the SIR model, those are the parameters you would use. Thus, one would then have all the analytical methods for the SIR model at your disposal.

2. C-SAM

- a. Support Intelligence, a DHS contractor and experts in security telemetry, devised a metric for evaluating a network's security called C-SAM (Cyber Security Abuse Metric). This single number quantifies an organization's strength in their information security practice and allows for easy comparison between two networks or network simulations. The C-SAM algorithm is proprietary, though Support Intelligence has expressed interest in licensing it to the SENDS effort.

2.4 Utilities

2.4.1 Simulation animation

The SENDSim animation displays simulation events and tracks key metrics over time.

2.4.2 Save/load scenarios

Users will have the ability to save their set of input levers and assumptions as a single file to share or load later.

2.4.3 Compare scenarios

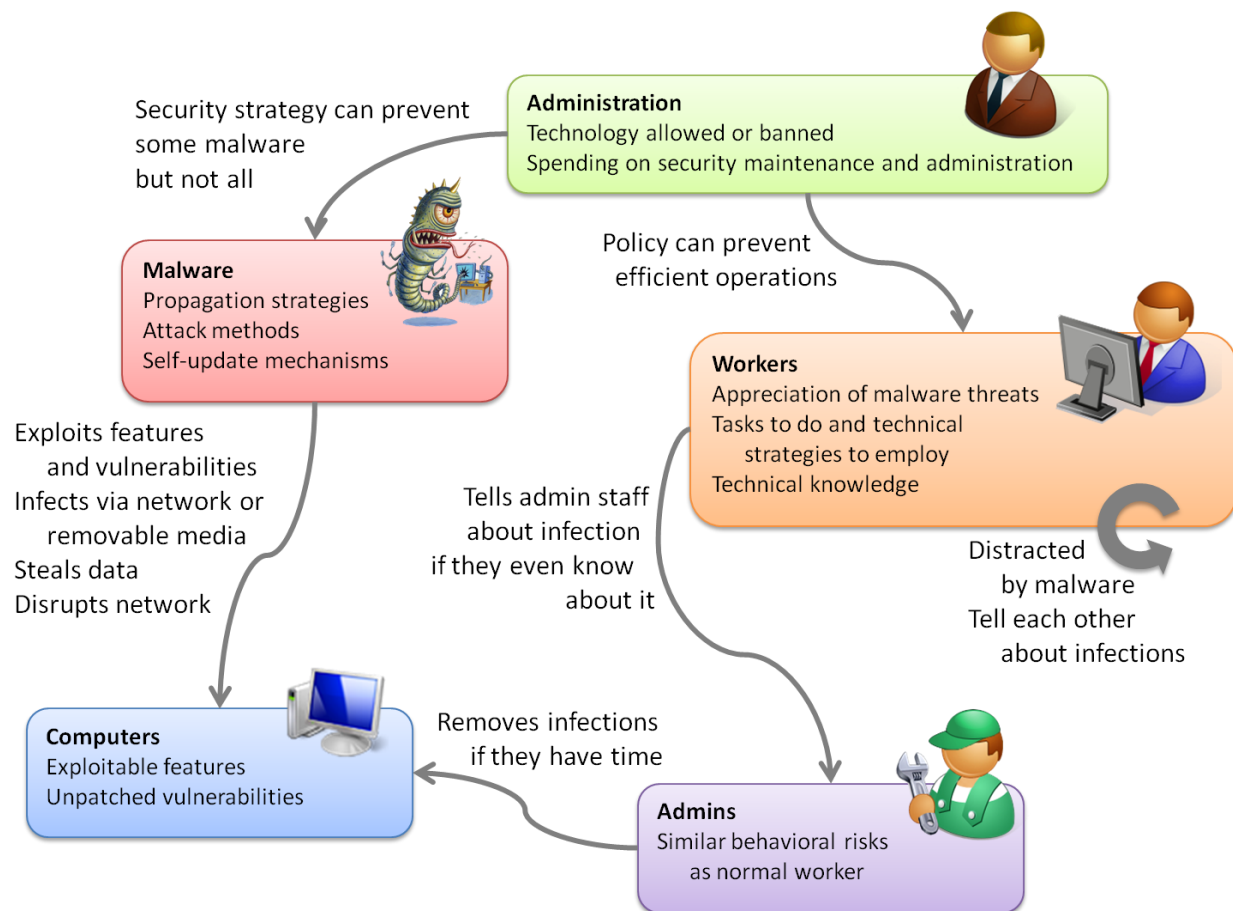
Users can compare scenario results by running them sequentially. Each new simulation result will draw a new

sparkline in the animation and present a new set of results in the analysis view.



3 Model

This section describes components related to the modeling and simulation, not necessarily how it is used by or exposed to the user.



3.1 Computer network

3.1.1 Connectivity

1. The computer network uses a simplified topology, as SMEs have indicated that more complex topologies won't necessarily offer any more insight into cybersecurity. Workstations are organized into subnets (corresponding to the social networks workgroups), connected to the larger network through switches. Each subnet's switch is then connected to the rest of the network via a main switch, which in turn is connected to the Internet via a firewall.
2. At the workstation, switch, and firewall levels, traffic may be blocked by configuration. E.g., the network may be configured to allow NetBIOS traffic within a subnet but not across subnets or across subnets but not over the Internet.
3. Filters to detect and filter malware-patterned traffic. May reduce malware spread but also slow network.

3.1.2 Initialization

1. The user will specify the number of workstations, the average subnet size, and the standard deviation of the subnet size.
2. The network will be initialized with the specified number of workstations and clustered into subnets of random size, normally distributed around the mean and standard deviation specified by the user.

3.2 Malware

3.2.1 General

SENDSim uses an abstraction of malware behavior, such that new (and even not-yet-existing) malware can be modeled easily. This malware abstraction involves:

1. Autonomous propagation via a physical network
2. User-mediated propagation via the use of removable media, email, or other actions by a human.
3. Discrete computer network vulnerabilities exploited by the specific worm or virus and required for their spread.
4. Discrete computer network features required to be enabled for the malware to spread.
5. Attack mechanisms employed once malware has infected a machine.

Conficker and Stuxnet have been implemented as proof that this abstraction can be applied to real-world, important malware threats.

3.2.2 Autonomous propagation

Malware, once it has infected a computer, may attempt to propagate to other computers without requiring any user action. The general algorithm is:

1. Scan the network to find other computers reachable via a specified protocol.

2. For each reachable computer (target):
 - 2.1. If the target meets all of the feature and vulnerability requirements of one of the malware's infection mechanisms, infect the target machine and execute the payload actions.
3. Schedule another propagation attempt after a delay.

For example, in order for Conficker to propagate to a target computer, NetBIOS traffic (port 445) must be allowed between the source and the target, and either the NetBIOS buffer overrun vulnerability (MS08-067) or an ADMIN\$ share with a poor password must exist on the target. If either of these vulnerabilities exist, then Conficker will deliver the following infection payload:

1. Remove buffer overrun vulnerability.
2. Disable autoupdate.
3. Disable antivirus software.
4. Immediately schedule propagation from the target computer.
5. Add listener to copy an autorun version of itself to any inserted removable media.
6. Initiate attack commands. (See "3.2.4 Attack mechanisms" below.)
7. Rerun propagation from the source node after a 30 minute delay. (See [http://mtc.sri.com/Conficker/.](http://mtc.sri.com/Conficker/))

Stuxnet applies a similar strategy, adding the exploitation of three more vulnerabilities (Printer Spooler Service MS10-061; LNK/PIF MS10-046; PLC Rootkit) and a new attack mechanism for destroying SCADA-controlled hardware.

3.2.3 User-mediated propagation

Some propagation mechanisms require explicit user action. For example:

1. Removable media / AutoRun
 - a. When a user inserts writable removable media into an infected computer, some malware may copy itself onto that media.
 - b. When a user inserts infected removable media into a computer, some malware may infect the computer. Some, like Conficker, may require that the AutoRun feature be enabled.
2. File insertion
 - a. Some malware may insert themselves into files of a particular type.
 - b. When that file is shared with someone (via email attachment or other mechanisms) and opened on a new computer, that computer may become infected.

3.2.4 Attack mechanisms

1. Physical damage
 - a. Once Stuxnet reaches a computer with a vulnerable SCADA device attached, it will send control instructions that physically damage the connected hardware. This increments the cost metric according to the physical damage cost specified by the user.
2. Data exfiltration
 - a. Every workstation will be marked as having some level of sensitive information on it. Some malware will send that data from an infected computer out to the internet.
 - b. For a first pass, we may simply assume that if a computer is infected with data-exfiltration-capable malware, its data is compromised, and that event will register in the information-assurance metric.
 - c. Later passes may model the actual exfiltration:
 - i. Whether it's encrypted.
 - ii. What port or protocol is used
 - iii. Whether a single IP address or dns name is set as the destination.
 - d. These would open up opportunities to model countermeasures, such as blocking particular packet patterns at the firewall.
3. Denial-of-service
 - a. Some malware may disable a network by flooding it with traffic, a so-called denial-of-service attack.
 - b. The effect of such an attack on worker productivity will increase as more infected computers participate in the attack.
4. Disable workstation
 - a. Some malware may prevent any legitimate use of the infected computer.
 - b. Worker productivity will decrease as they are forced to use task strategies that don't involve the infected computer.
 - c. The computer will require repair, either by the computer's user (if they're sophisticated enough) or an admin.

3.3 Computer

3.3.1 General

Computers have vulnerabilities and features that may be exploited by malware. However, some features are also required to permit worker agents to successfully and efficiently perform their daily work tasks.

3.3.2 Auto-update

1. Computers set for auto-update should pull updates at specific interval, which may or may not require a restart.
2. Find out what schedule MS sets for auto-updates by default.
3. Let user specify interval and whether or not all workstations are auto-updated at the same time or whether their times are randomized.
4. When an update is occurring on a workstation, it may prevent or delay a user from accomplishing tasks, at which point the user may cancel the update or disable auto-update in total.

3.3.3 Anti-virus

1. Some computers will have anti-virus software installed.
2. AV software may be run at will by the user or on a schedule.
3. When run, the software will remove any malware for which it has a matching rule.
4. As with auto-update above, scans may prevent a user from accomplishing their tasks, and they may respond by cancelling the scan or cancelling auto-scan in total.

3.3.4 SCADA

1. Some computers have SCADA (supervisory control and data acquisition) control systems attached to them. These systems typically control physical systems, such as a Uranium centrifuge.
2. Computers with SCADA systems may be vulnerable to damage to the physical system if infected with Stuxnet.

3.4 Worker

3.4.1 Social Network

Not every worker knows and communicates with every other worker. Instead, there is a social network wherein most workers have close links with their workgroup as well as some links outside their workgroup. We approximate this phenomenon using a modification of the [small-world](#) social network construction methodology:

1. There is a one-to-one correspondence between workgroups and subnets. That is, a subnet with N computers also has N workers who all have social network links to each other.

2. In addition to intra-workgroup links, workers may have links to random workers outside their workgroup. The number of extra-workgroup links is determined randomly with the distribution roughly matching a power law. Specifically, $n = mr^p$, where n is the number of links for the worker, m is the maximum number of links, r is a random number between 0 and 1, $p = 100^{1-c}$, and c is the user-entered connectivity parameter, between 0 and 1.
3. A worker's extra-workgroup connections are likely the result of moving from one workgroup to another. Thus, we may want the probability of connecting worker a to worker b to be partly a function of whether worker a is also connected to other workers in b 's workgroup.
4. If and when we model the concept of newly hired workers, we may want to initial connect them to only their workgroup and have extra-workgroup connections grow over time.

3.4.2 Computer use

Each worker is associated with one workstation computer.

3.4.3 Threat appreciation

["Availability heuristic"](#)—notion that people tend to estimate the risk/frequency/prevalence of a phenomenon based on the ease with which they can come up with an example. When a coworker tells you about an infection, you're more likely to appreciate the threat (might even *overestimate* the risk) because of the availability heuristic.

The threat appreciation model is inspired by marketing research conducted on product perception. Data demonstrate that perception follows three trends:

1. Increases in perception follow a *law of diminishing returns*. E.g., you can't max out someone's perception of a product by showing them a TV ad a thousand times. That thousandth viewing has much less of an impact than the first. Likewise, the thousandth person to tell me about a malware threat is not likely to have a large impact.
2. These perceptions do not remain at these high states, but rather there is a *passive decay back to a baseline* over time. E.g., a TV ad may increase your perception of a product, but eventually you forget about the ad and your perception decreases. Each worker will be initialized to a random baseline threat appreciation.
3. Research demonstrates that the rate of decay is related to the *channel* from which the perception was impacted. E.g., a person's perception of a product based on their own use of the product will decay much more slowly than perception built on viewing a TV ad.

Similarly, we may find that a user's threat appreciation may decay much more slowly after suffering a malware infection themselves than from hearing about threats during a cybersecurity education seminar.

3.4.4 Discover infection

A worker may discover an infection on their computer the next time they attempt a task on that computer after an infection. At that point, they may:

1. Tell one or more coworkers about the infection.
2. Attempt to clean the infection on their own.
3. Tell an admin about the infection.

3.4.5 Tell co-worker about infection

1. Within an hour or so of discovering an infection, a worker will tell one or more coworkers about the infection.
2. This will raise the threat appreciation of the coworker.

3.4.6 Tell co-worker about fix

1. Within an hour or so of fixing an infection, a worker or admin may tell one or more coworkers about the fix.
2. This will raise the technical knowledge of the coworker.

3.4.7 Power workstation on/off

1. Some workers may turn off their computer at the end of the day and turn it on in the morning.
2. Since malware can only infect a computer when it's on, we may find that malware propagations that appeared to stop one day may start up again on the next workday, when vulnerable computers are turned on.

3.4.8 Enable/disable feature

1. In an attempt to complete work tasks, workers may enable or disable computer features.

2. For example, a worker may turn on windows file sharing to share a file or turn off auto-update because it was trying to patch the OS while the worker was attempting a task.

3.4.9 Apply patches and updates

3.4.10 Attempt work task

1. In order to do their jobs effectively, workers need to accomplish a set of tasks each day.
2. Each task has a number of strategies that may be employed to accomplish the task.
3. Workers will attempt to accomplish a task by first employing the strategy that has the lowest perceived effort and the lowest risk of failure. The perceived effort includes the effort to learn the strategy, if it requires more technical knowledge than the worker has. If that first strategy fails (e.g., because of security constraints), they will move on to the next best strategy.
4. If a worker accomplishes a task, that contributes to the productivity metric. If they exhaust all possible strategies for a task, then they move on to the next task without contributing to productivity.
5. E.g., a worker may have a task of sharing a file with a coworker. Their preferred method may be an email attachment, and if the email server doesn't support the attachment size, they may resort to removable media or personal email.
6. Each strategy may have security consequences.
7. Possible tasks and strategies:

Task	Strategies
1. Send a file	Email attachment; file server; file sharing; removable media; personal email - Some of which then need to delegate to ShortDialog
2. Short dialog	IM; email; phone call; in person
3. Long narrative	Email; delegate to SendFile
4. Teleconference	Phone call + delegate to SendFile; Install software

5.	Access website	From workstation; from home
6.	Install software	On workstation; at home + delegate to SendFile
7.	Remote access	VPN; from home + delegate to SendFile
8.	Download content	To workstation; To home + delegate to SendFile
9.	Data access	Direct access; if fails for lack of permissions, ask an admin for permissions (with delay)

Some strategies may have three forms:

- Attempt the strategy without any change to workstation/network configuration
- Attempt the strategy, with the worker making changes to the config themselves
- Attempt the strategy, and ask an IT admin to change the config

3.4.11 Choose password

Workers will choose a password:

- When they first join the network.
- When prompted by an admin or by the authentication system.
- When they hear about a password-related security event.

3.4.12 Insert removable media

Workers may insert removable media into a computer as part of a task strategy.

3.4.13 Bring work laptop home or personal laptop into work

- Either may be used by a third party, e.g., worker's teenager

3.4.14 Create ad hoc network

3.4.15 Training and education

3.5 IT administrators

3.5.1 (Remotely) detect infection

3.5.2 Fix workstation

When a worker discovers an infection, they may inform the IT department about the infection. The IT department will then dispatch an administrator to disinfect the workstation. Delays may occur if all IT staff are already busy.

3.5.3 Educate worker

3.5.4 Hold training classes

3.5.5 Disconnect subnet

IT admins may apply different strategies for slowing the spread of an infection, such as disconnecting an infected subnet from the rest of the network.

3.6 Vendor

3.6.1 Publish patch/update

3.7 Hacker

3.7.1 Publish new exploit (zero-day)

3.8 Insider threat

An insider threat agent inherits properties from a normal worker agent, only their task queue is also filled with data exfiltration tasks. An insider threat may act alone or in concert with a network of insiders. They may all have data exfiltration tasks, but they may have so few that, individually, they don't trigger any alarms.

4 Example questions and metrics

The following are some example questions that might be answered through use of the simulation tool. As we explore our sources of data and research, the precise scope of the tool's utility will evolve, but the general vision is to provide the right input levers and the output metrics for users to make actionable conclusions about network security policy.

1. *How can a change in policy (enforced by HR or enforced by technology) increase network security without decreasing user productivity?*

Clearly, the most secure network is the one that is turned off, but then no one can get anything done. Network design and policy can both enable and impede users' work tasks and, thus, directly influence their productivity. By modeling both the human and technical sides of network security, the tool will enable policy makers and analysts to take a broader view of network security and account for larger institutional goals beyond IT. Model input policy levers might include: password quality checks, administrator privileges granted/denied to end users; user task technical requirements; disabling USB ports and DVD drives on workstations; instructing end users not to install unauthorized software; etc. Output metrics might include: total worker productivity¹; average worker frustration²;

¹ Worker productivity is notoriously difficult to quantify for any position that doesn't involve manual, repetitive labor. However, it is possible to measure productivity changes from a baseline due to a specific influence. For example, the prototype may model users as having specific tasks to accomplish in order to do their job, and each task might require certain computing and network resources. If there are no network impediments to the user accomplishing their tasks, we can say that they are 100% productive. If a malware attack or an overzealous security policy prevents them from accomplishing a task that is 10% of their job, we could say they are 90% productive. While such a measure may not have a dollars-and-cents precision, it can be very helpful for qualitative comparisons of different policies and strategies.

² Worker frustration alone is not an output metric of ultimate concern, but tracking the number of worker frustration events, such as trying to accomplish a computing task and failing because of network outages or strict IT policies, can provide deeper insight into user behavior as it relates to network security. More frustrated workers are more likely to subvert security policies and thus increase overall network vulnerability.

number of Conficker-vulnerable workstations; number of Conficker-infected workstations; etc.

2. *Which solution results in a better outcome, expanding the IT security and administration staff or educating and empowering end-users?*

Both before and during a Conficker infection, having the right people with the right training is critical. The right mix, however, is not immediately intuitive. Will it make for a more secure network to double your IT staff so they can protect and disinfect workstations faster, or would it ultimately be better to apply those resources toward educating end users, so they can protect or fix their own workstations? Agent-based modeling can simulate the interactions between IT admin, end-user, and malware “agents” to demonstrate how well a network of people can defend against a network of Conficker infections. Outcomes of these interactions, including counterintuitive outcomes, then emerge from these simulated interactions. Input levers might include: the size of the IT staff; the baseline level of end-user computer skill; computer skill education policy; etc. Output metrics might include: the amount of time it takes to eradicate a network infection after its initial discovery; worker productivity before and during an attack; etc.

3. *What does the timescale of a Conficker infection look like given my particular network and user profiles? What aspects of my users and network policy are enabling or counteracting the spread?*

In addition to proscriptive questions like those listed above, the simulation can be used as a descriptive tool for better understanding the current network risks and opportunities. Input levers might include: workstation default configurations; user permissions; firewall settings; soft and hard IT policy; etc. Output metrics might include: vulnerabilities and infections over time; short-term and long-term time series of infection propagation; etc;

4. *How can a real malware infection ultimately strengthen network security by improving threat perception and awareness amongst end users?*

The human brain has evolved to appreciate physical threats, the kind a caveman might encounter. We are therefore predisposed to think of threats that are physically far away to be less important and for attacks that happen directly to us or to people we know as being more real than reports of attacks that happen elsewhere. Such a disconnect between human threat perception and the physically agnostic threats of cyberspace can lead to counterintuitive outcomes of malware attacks. When a user suffers a malware attack or hears about an attack from a coworker, their perception of malware threats generally is greatly increased. Thus, after an attack, your workforce may evolve a better culture of security than could ever be created by education or policy alone. By modeling users’ threat

perception and the change in that perception after suffering an attack or hearing about nearby attacks by word-of-mouth, the tool can simulate both the short-term negative and long-term positive impacts of malware events. Input levers may include: user social connectivity; initial network vulnerabilities; initial threat perceptions; etc. Output metrics may include: threat perception over time; network vulnerabilities and infections over time; etc.

5. *How might my staff react to combat a “zero day” Conficker attack? How would network functionality and worker productivity change, and hopefully rebound, over time?*

It is theoretically possible to completely protect your network from Conficker’s known infection mechanisms. However, it is possible—even likely—that future variants of Conficker will exploit some vulnerability for which a patch does not yet exist. Understanding how such a “zero day” attack would affect worker productivity and how well your staff can respond to the attack is both useful and not necessarily obvious. Model input levers might include: number of days until a patch is available, fraction of workstations that are vulnerable, etc. Output metrics might include: timeseries of infection density and worker productivity before the attack, during the attack, and after a patch is available; etc.

6. *Should users be permitted to bring personal laptops into work or bring work laptops home? How effective is a policy banning such practices?*

No matter how locked-down your network, firewall, and workstations are, threats can permeate the bounds of the network when hardware is physically moved between the network and the outside world. However, a policy banning such practices may be difficult to enforce without significant investment. For example, secure facilities for conducting classified research make it very difficult to bring in outside hardware, but they require a full-time security staff, metal detectors, and a thorough search of personnel. Such a ban might not be the best solution: worker productivity and job satisfaction may depend on a flexible work environment, such as the ability to access work resources when and where the user wants to. A hard-nosed approach to network security may ultimately lead to less effective workgroups. Model input levers might include: hardware portability policy {permit, deny}; enforcement {written policy only, inspections at facility entrance}; etc. Output metrics might include: worker productivity (baseline and during a Conficker attack); security expenditure (absolute and relative to productivity gains and losses); network security (density of vulnerabilities and infections); etc.

7. *What combination of policy and network design will help me meet my security and productivity goals?*

Given all the possible combination of security policies and network designs, as well as the sometimes counterintuitive impact they can have on worker productivity, it can be difficult to find the best blend of flexibility and security to support the healthy operation of your workgroups. The optimizer will use non-linear techniques such as evolutionary algorithms to explore the solution space to find the policy and design that maximizes long-term worker productivity subject to minimum security and short-term productivity constraints. Model input levers might include: minimum level of data security required; policy and network configuration types to consider; minimum short-term productivity permitted; etc. Output metrics might include: recommended policies and configurations; timeseries of worker productivity, at baseline and during a Conficker attack, given these recommendations; data security given these recommendations; etc.